

**DEPARTAMENTO DE ELECTRÓNICA Y AUTOMÁTICA**

**FACULTAD DE INGENIERÍA – UNIVERSIDAD NACIONAL DE SAN JUAN**

Informe de Practica Nº3

CONTROLADORES DE PARÁMETROS DISCRETIZADOS

**Asignatura:** CONTROL 3

**Ingeniería Electrónica**

***Autores (Grupo Nº 4):***

*Albornoz Rubén Fernando - Registro 9827*

*Avila Juan Agustín - Registro 26076*

**1º Semestre**

**Año 2020**

# Planta Hidráulica.

Implemente un controlador PID con aproximación rectangular para la siguiente planta:

|  |  |  |
| --- | --- | --- |
|  |  | (1) |

## Determine el período de muestreo y justifique la elección del mismo.

Primero se obtienen los polos de G(s). Para ello, se utiliza el siguiente comando de matlab:

%% punto 1.2

p=pole(G) %Se obtienen los polos de la planta

Obteniendo el siguiente resultado:

p =

-2

-1

Por lo tanto, la menor constante de tiempo será dada por el polo que está mas alejado del origen, es decir:

Por el teorema de Shannon se sabe que el tiempo máximo de muestreo no puede ser mayor a 0,5 Tmin, en este caso se elige un valor de muestreo cinco veces menor al Tmin:

## Explicite la función de transferencia discreta en forma analítica. Emplear el comando ‘c2dm’ de Matlab para comprobar el valor obtenido.

Para obtener la transformada z por convolución, se utiliza la siguiente formula:

|  |  |  |
| --- | --- | --- |
|  |  | (2) |

Al utilizarse un retenedor de orden cero, se debe multiplicar la ecuación anterior por la función de transferencia del ZOH:

|  |  |  |
| --- | --- | --- |
|  |  | (3) |

Luego, se calculan los residuos usando la sig. fórmula:

|  |  |  |
| --- | --- | --- |
|  |  | (4) |

El segundo residuo será:

|  |  |  |
| --- | --- | --- |
|  |  | (5) |

El tercer residuo será:

|  |  |  |
| --- | --- | --- |
|  |  | (6) |

*Reemplazando los valores de las ecuaciones (4) (5) y (6) en la ecuación (2), se tiene:*

Normalizando, queda:

|  |  |  |
| --- | --- | --- |
|  |  | (7) |

Para comprobar en matlab:

[nd,dd]=c2dm(n,d,T0,'zoh')

Gz=c2d(G,T0,'zoh')

nd = 0 0.0045 0.0041

dd = 1.0000 -1.7236 0.7408

Gz = 0.004528 z + 0.004097

-----------------------------

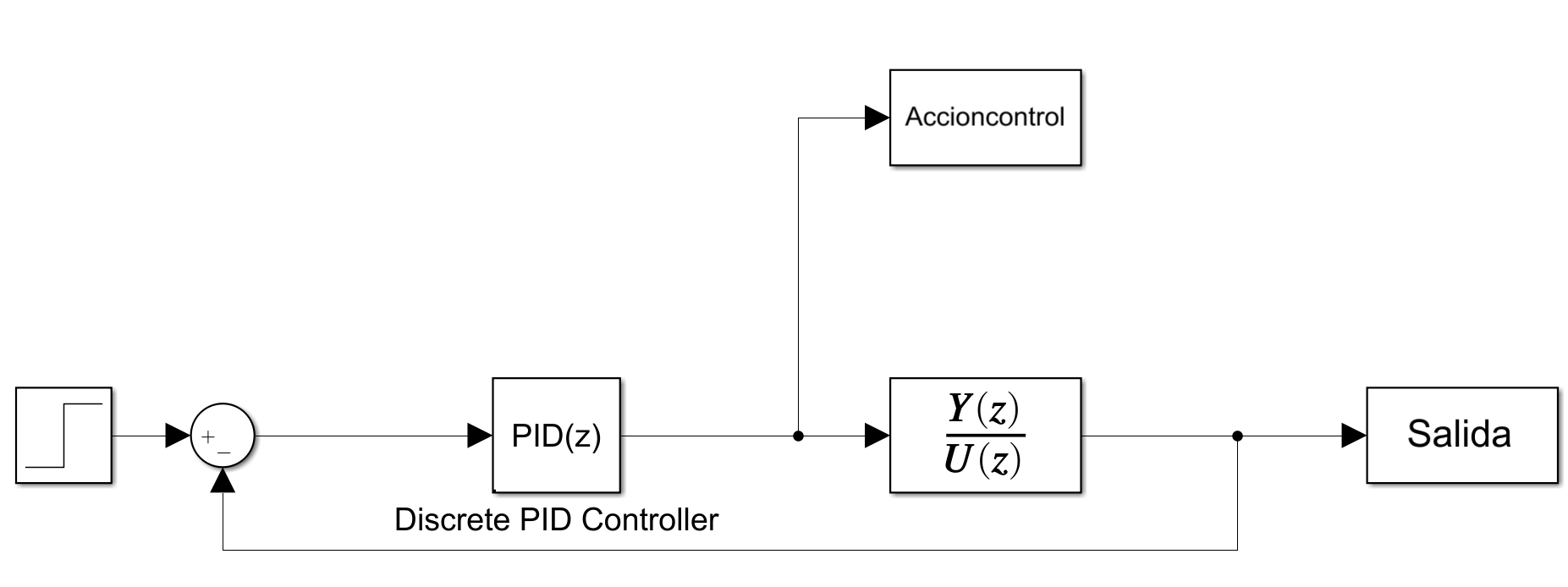
z^2 - 1.724 z + 0.7408

Sample time: 0.1 seconds

Discrete-time transfer function.

## Calcule los parámetros del controlador empleando prueba y error de la siguiente manera: emplee gráficas superpuestas de Matlab para mostrar el efecto de cambiar cada uno de los términos comenzando por el P, luego el D y finalmente el I (con al menos 5 valores de cada uno e indicando la tendencia de aumento de la constante).

Para esto, se armó la siguiente planta en simulink:



Dentro del controlador PID, se declararon kp,ki y kd como variables del entorno de trabajo, y lo mismo se realizó para la función de transferencia discreta, definiendo al numerador y denominador como las variables obtenidas del comando c2dm. Además, en todos los bloques el tiempo de muestreo se definio con la variable “T0”.

Se utilizó el siguiente código de matlab:

%% punto 1.3

ki=0;kd=0; %define kd y ki=0 para variar solo kp

mkp=2; %los m son los "pasos" que dara en cada prueba

nkp=2; %para fijar una constante, se elige m\*n

mkd=2; %En este caso, kp=mkp\*nkp=2\*2=4, kd=2\*1=2

nkd=1; %se usaron estas variables para facilitar

mki=1; %el uso de distintos rangos

figure();

Legend=cell(5,2);% Arreglo de celdas para nombrar señales en la grafica

for i=1:5 %hace 5 iteraciones, desde 1 hasta 5

kp=mkp\*i; %kp varia desde mkp\*1 hasta mkp\*5

sim('punto1.slx'); %simula la planta en simulink

subplot(2,1,1); %grafica la salida en la parte superior

plot(Salida);

Legend{i,1}="Salida con Kp="+kp;

hold on;

subplot(2,1,2);

plot(Accioncontrol);%Grafica la accion de control

hold on;

Legend{i,2}="Accion con Kp="+kp;

end

subplot(2,1,1); %fuera del bucle, se vuelve a la grafica superior

title('Salida de la planta variando el Kp');

legend(Legend{:,1});grid on; %agrega todos los nombres de las señales

subplot(2,1,2);

title('Accion del controlador variando el Kp');

legend(Legend{:,2});grid on;

%% para kd

kp=mkp\*nkp; %Fija el Kp en un valor dado por m\*n

figure();

Legend=cell(6,2);

for i=0:5 %En este caso se comienza el bucle desde cero para

kd=mkd\*i; %graficar la respuesta sin la accion derivativa

sim('punto1.slx');

subplot(2,1,1);

plot(Salida);

Legend{i+1,1}="Salida con Kd="+kd;

hold on;

subplot(2,1,2);

plot(Accioncontrol);

hold on;

Legend{i+1,2}="Accion con Kd="+kd;

end

subplot(2,1,1);

title('Salida de la planta variando el Kd');

legend(Legend{:,1});grid on;

subplot(2,1,2);

title('Accion del controlador variando el Kd');

legend(Legend{:,2});grid on;

%% para ki (procedimiento igual a kd)

kd=mkd\*nkd;

figure();

Legend=cell(6,2);

for i=0:5

ki=mki\*i;

sim('punto1.slx');

subplot(2,1,1);

plot(Salida);

Legend{i+1,1}="Salida con Ki="+ki;

hold on;

subplot(2,1,2);

plot(Accioncontrol);

hold on;

Legend{i+1,2}="Accion con Ki="+ki;

end

subplot(2,1,1);

title('Salida de la planta variando el Ki');

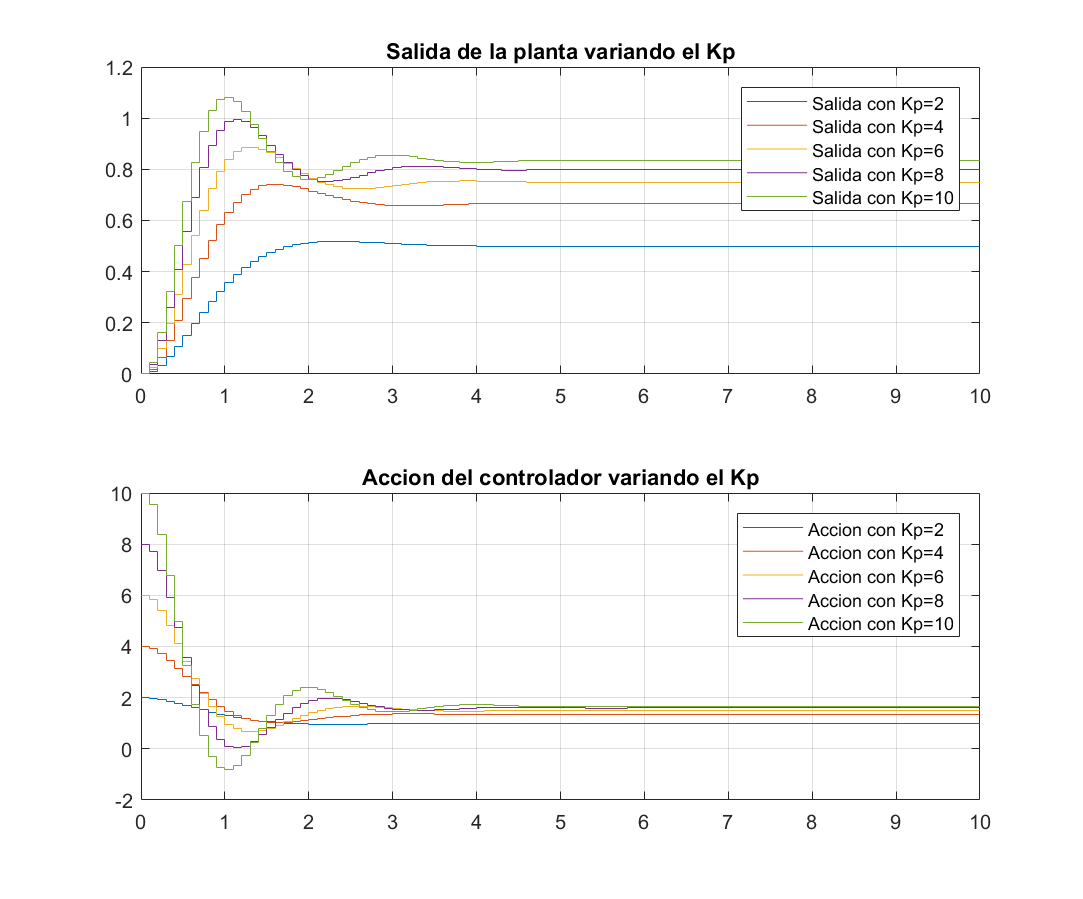
legend(Legend{:,1});grid on;

subplot(2,1,2);

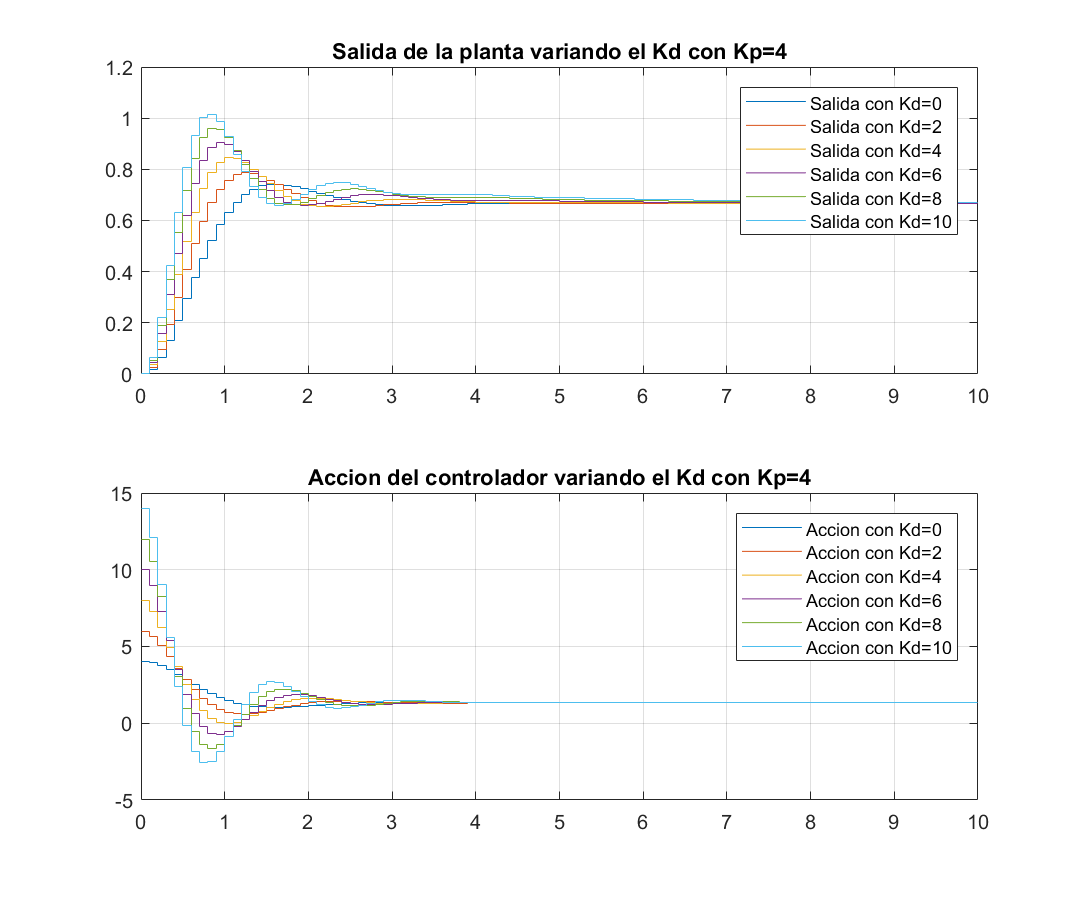
title('Accion del controlador variando el Ki');

legend(Legend{:,2});grid on;

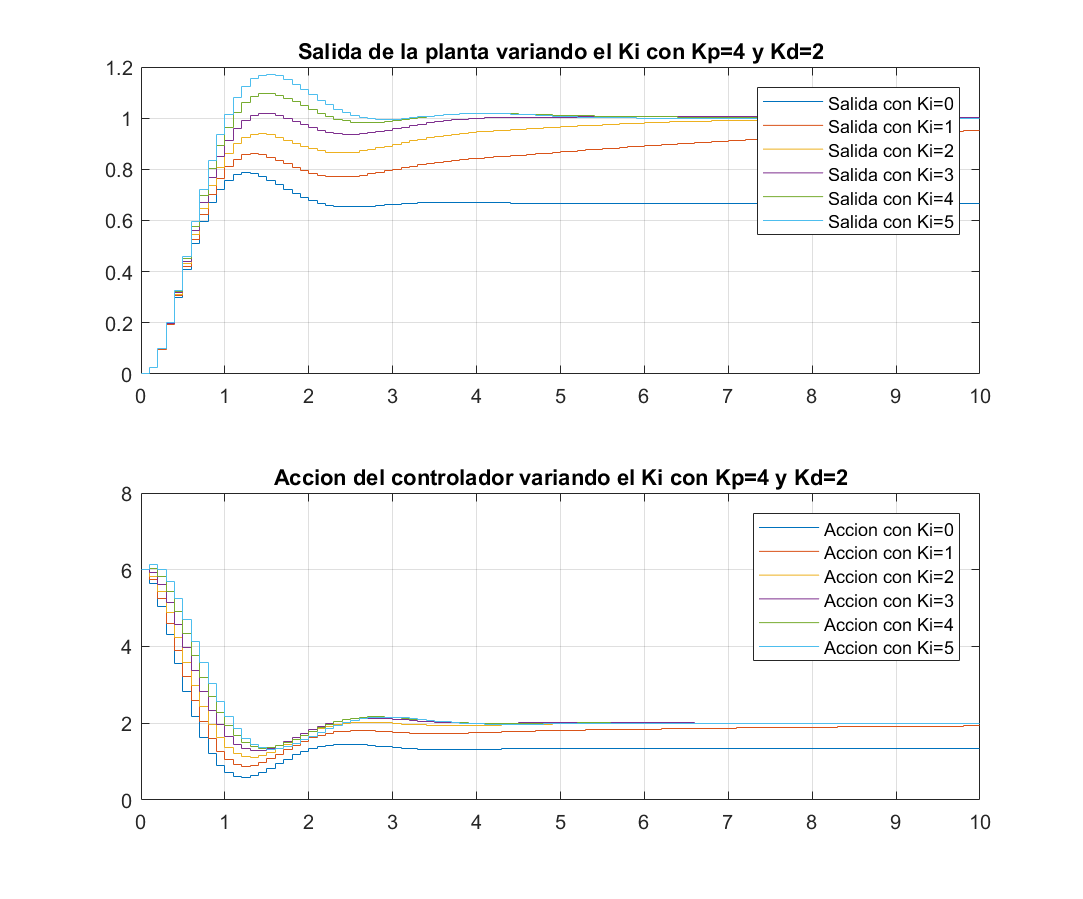
Las distintas salidas se reproducen a continuación:



En este caso, se observa que a mayor Kp, la respuesta del sistema a una entrada escalón es cada vez más rápida y el error en estado estacionario es menor, pero también a medida que se aumenta el Kp, aumenta el sobreimpulso y también aumenta la acción de control. En este caso, se eligió un Kp=4, ya que la respuesta no es tan lenta, el sobreimpulso es bastante pequeño, la acción de control no toma un valor tan grande y

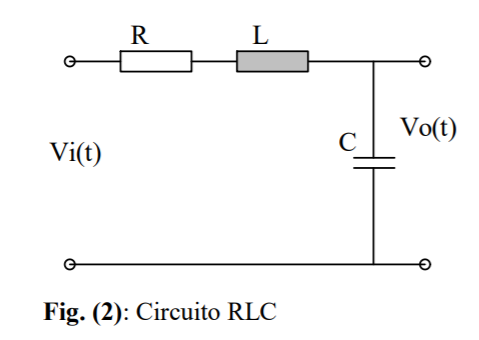


Se observa que al aumentar la ganancia de la parte derivativa,



(Descripcion de la acion integral)

# 2 PLANTA ELÉCTRICA DEL TP2. PID discreto.



Donde: R =10Ω L=10mHy C=10μF Vi(t)=10V

## 2.1 Determine el período de muestreo y justifique la elección del mismo.

Se sabe del TP2 que la función de transferencia de la planta eléctrica es la siguiente:

Los polos están ubicados en

1.0e+03 \*

-0.5000 + 3.1225i

-0.5000 - 3.1225i

Que es un par de polos complejos conjugados con parte real igual a -500. Por lo tanto su periodo es 1/500=2ms. Para cumplir con el teorema de Shannon, el tiempo de muestreo debe ser al menos la mitad del T de la planta, es decir, 1ms. En este caso, se selecciona una frecuencia de muestreo 5 veces mayor que la frecuencia de Nyquist, es decir:

|  |  |  |
| --- | --- | --- |
|  |  | (8) |

## 2.2 Calcule los parámetros del controlador mediante Ziegler-Nichols de lazo abierto. Muestre los resultados obtenidos en gráficas de salida para entrada escalón y luego para las acciones de control de manera que se puedan ver en forma simultánea, pero no solapada. (emplear subplot, y ubicar una debajo de la otra, que se pueda ver).

Se carga la planta en matlab y se analiza su respuesta ante una entrada escalon con la planta a lazo abierto:

%% punto 2

n=1;d=[.0000001 .0001 1];

H=tf(n,d);

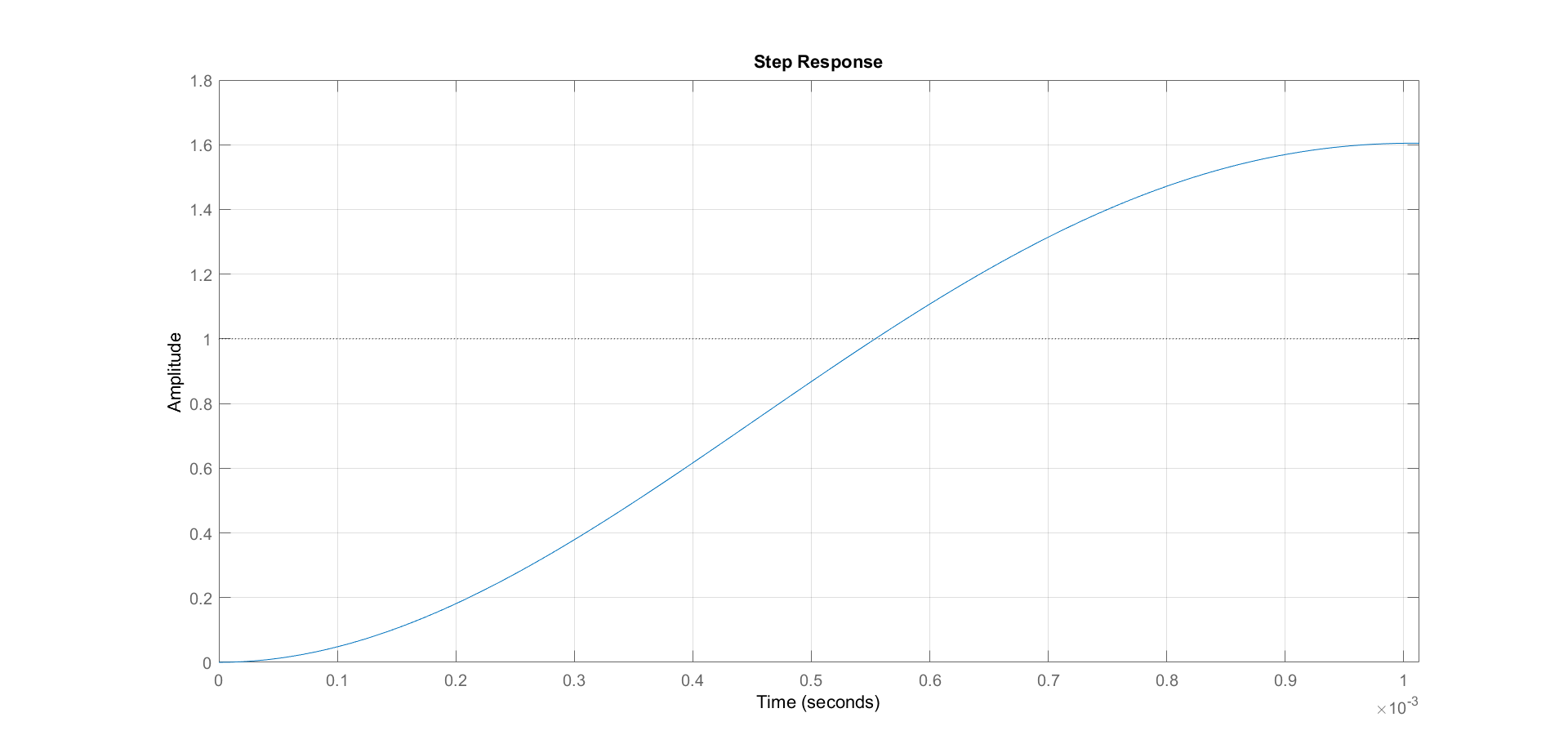
pp=pole(H);

T0=.0002;

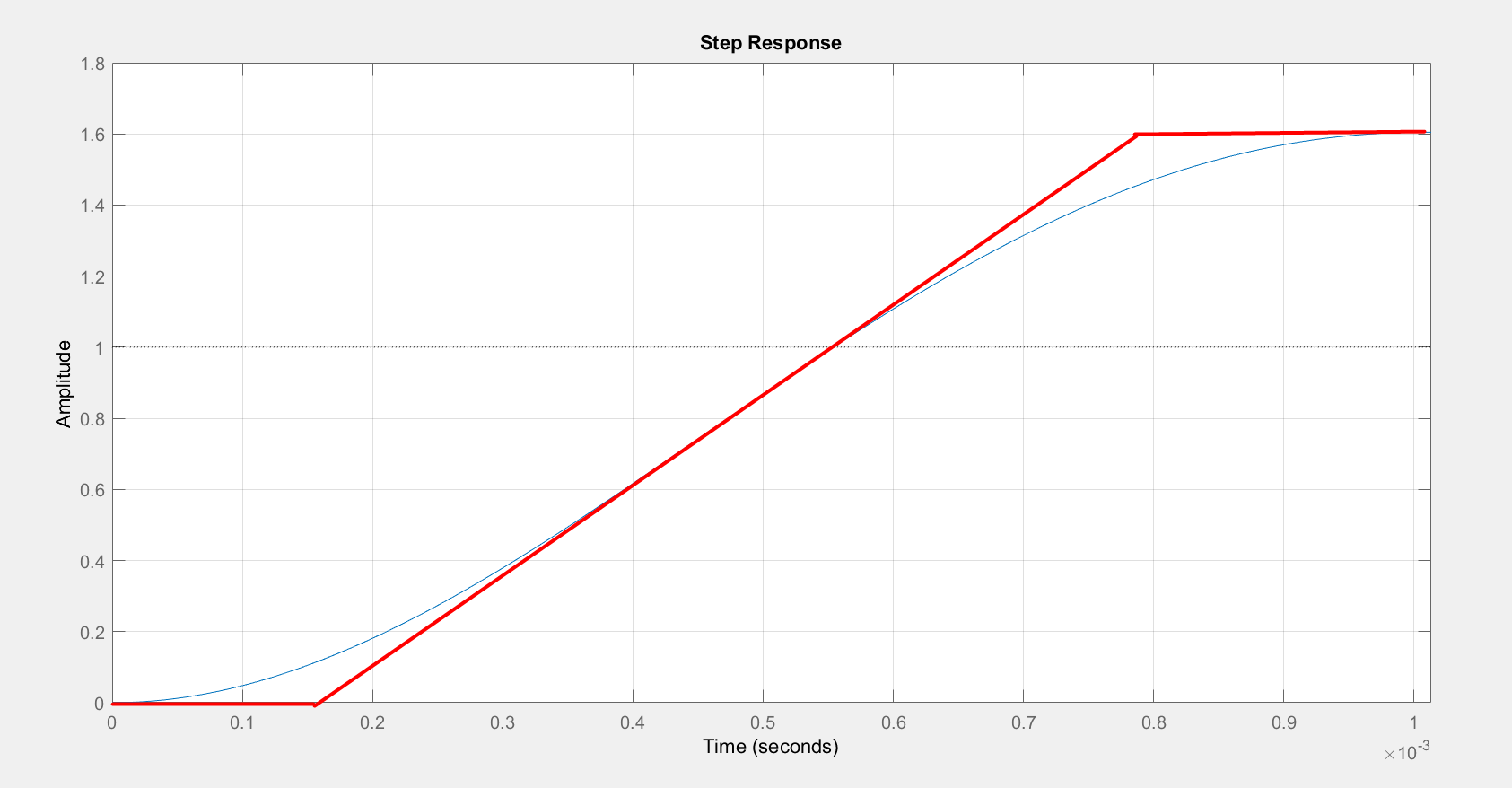
st=stepinfo(H)

step(H,st.PeakTime);grid

[nd,dd]=c2dm(n,d,T0,'zoh')



Haciendo una aproximación, se toman los siguientes valores:



Tu=0.00015

Td=0.0006

Ke=1.6

Y se desarrollan K, Ki y Td según las ecuaciones provistas en el apunte:

Tu=.00015;

Tg=.00075-Tu;

Ke=1.6;

K=((1.2\*Tg)/(Ke\*(Tu+(T0/2))))-((.3\*Tg\*T0)/(Ke\*(Tu+(T0/2))^2));

Ki=(.6\*Tg)/(K\*Ke\*(Tu+(T0/2))^2);

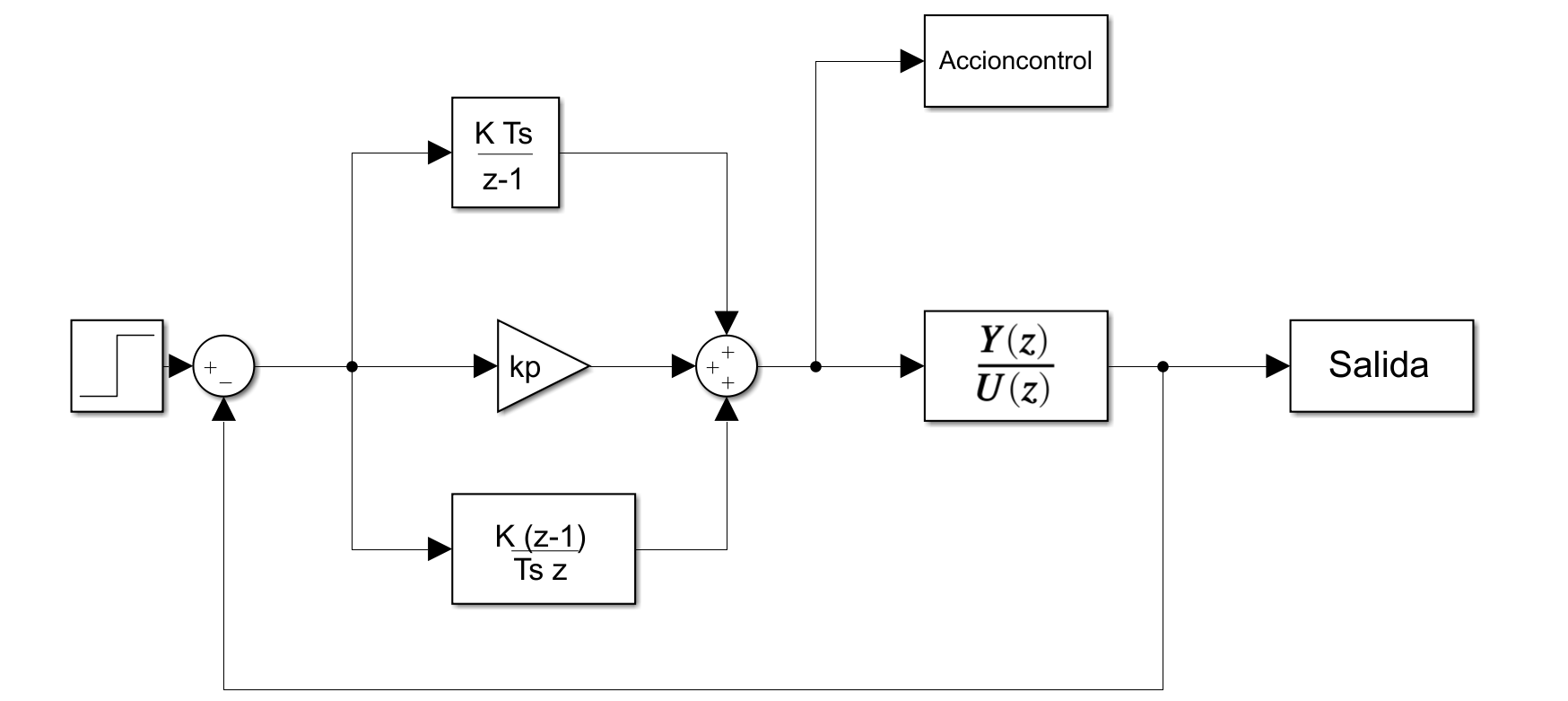
Td=.5\*Tg/(K\*Ke);

kp=K

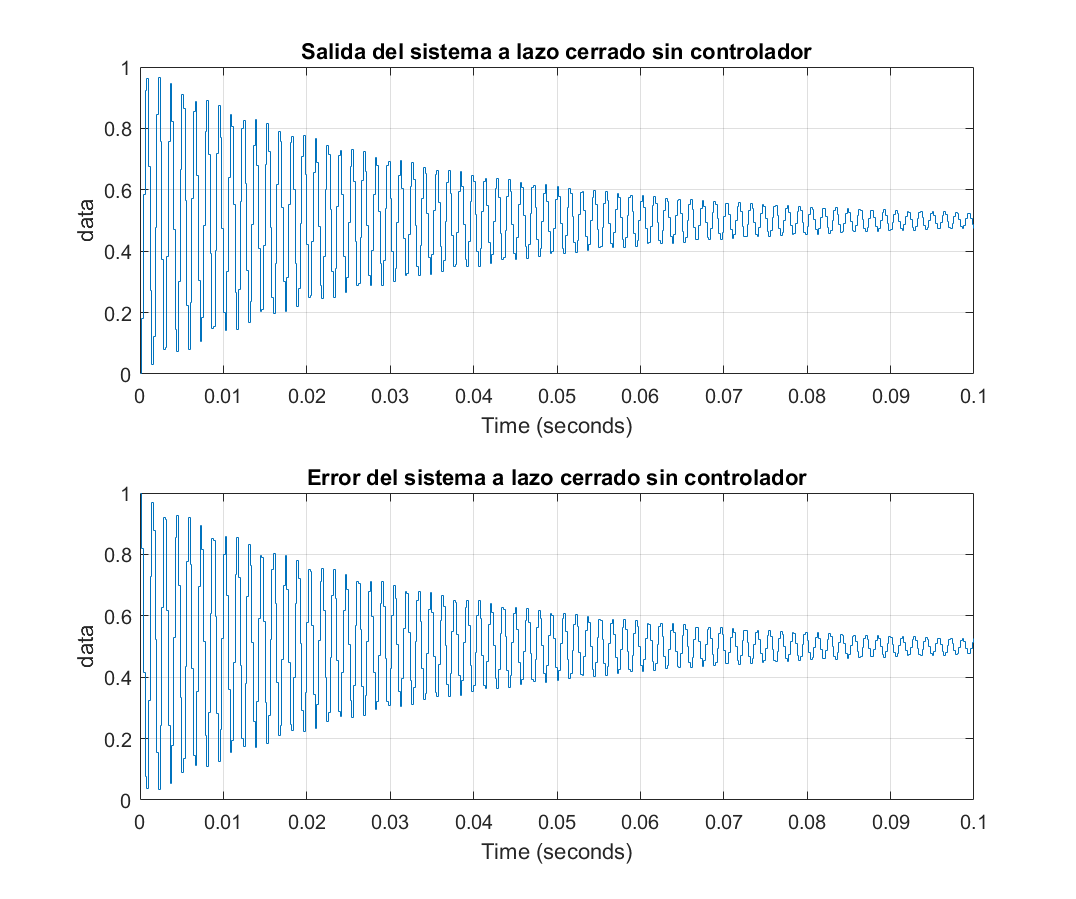
ki=K\*Ki

kd=K\*Td

Se arma la siguiente planta en simulink:



En primer instancia, se anulan los términos derivativos e integral para graficar la respuesta del sistema a lazo cerrado sin controlador y poder comparar:



En este caso, al ser Kp=1, la acción de control es igual al error.

Luego, se procede a simular el sistema con los valores calculados:

sim('punto2.slx');

figure();

subplot(2,1,1);

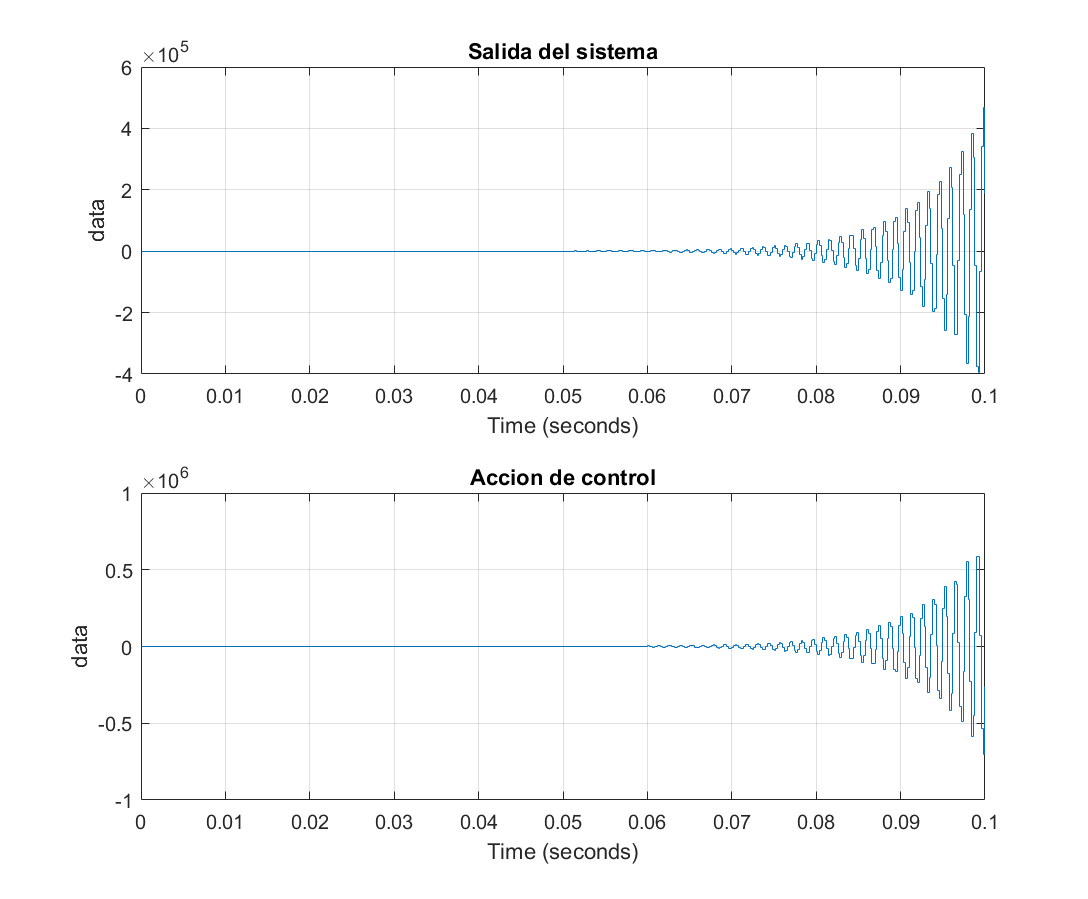
plot(Salida);

grid;title("Salida del sistema a lazo cerrado sin controlador");

subplot(2,1,2);

plot(Accioncontrol);

grid;title("Error del sistema a lazo cerrado sin controlador")



Los valores de ganancia son los siguientes:

kp = 1.4400

ki = 3.6000e+03 / ki\*T0= 0.7200

kd = 1.8750e-04 / kd/T0= 0.9375

Se observa que el sistema es inestable, por lo tanto se incrementa el valor de Tu a ver si los valores mejoran.

Tu=.0002;

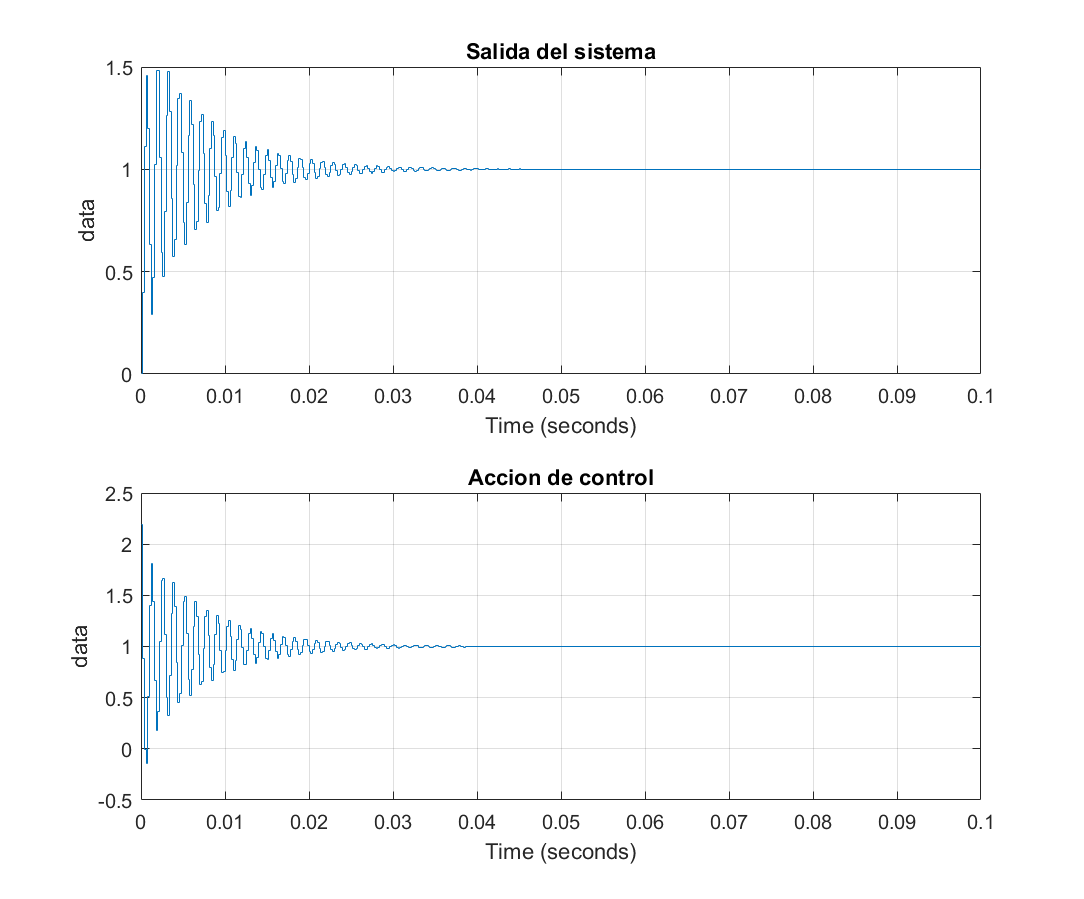
Tg=.0006;

Ke=1.6;

kp = 1.2500

ki = 2.5000e+03 / ki\*T0=0.5000

kd = 1.8750e-04 / kd/T0=0.9375



Se observa que el sistema ahora es estable y que la respuesta se estabiliza considerablemente más rápido que en el sistema sin controlador.

## 2.3 Recalcule por prueba y error hasta obtener un valor adecuado

Para la prueba y error, en vez de variar directamente las ganancias de cada acción de control, se varian los tiempos para ver como afecta esto al sistema. Suponiendo que el sistema es inestable para un Tu<0.0002, se cambian los tiempos para ver como afecta la salida:

figure();

Legend=cell(5,2);% Arreglo de celdas para nombrar señales en la grafica

for i=1:5 %hace 5 iteraciones, desde 1 hasta 5

Tu=.0002+i\*.00003

K=((1.2\*Tg)/(Ke\*(Tu+(T0/2))))-((.3\*Tg\*T0)/(Ke\*(Tu+(T0/2))^2));

Ki=(.6\*Tg)/(K\*Ke\*(Tu+(T0/2))^2);

Td=.5\*Tg/(K\*Ke);

kp=K

ki=K\*Ki

kd=K\*Td

sim('punto2.slx'); %simula la planta en simulink

subplot(2,1,1); %grafica la salida en la parte superior

plot(Salida);

Legend{i,1}="Salida con Tu="+Tu;

hold on;

subplot(2,1,2);

plot(Accioncontrol);%Grafica la accion de control

hold on;

Legend{i,2}="Accion con Tu="+Tu;

end

subplot(2,1,1); %fuera del bucle, se vuelve a la grafica superior

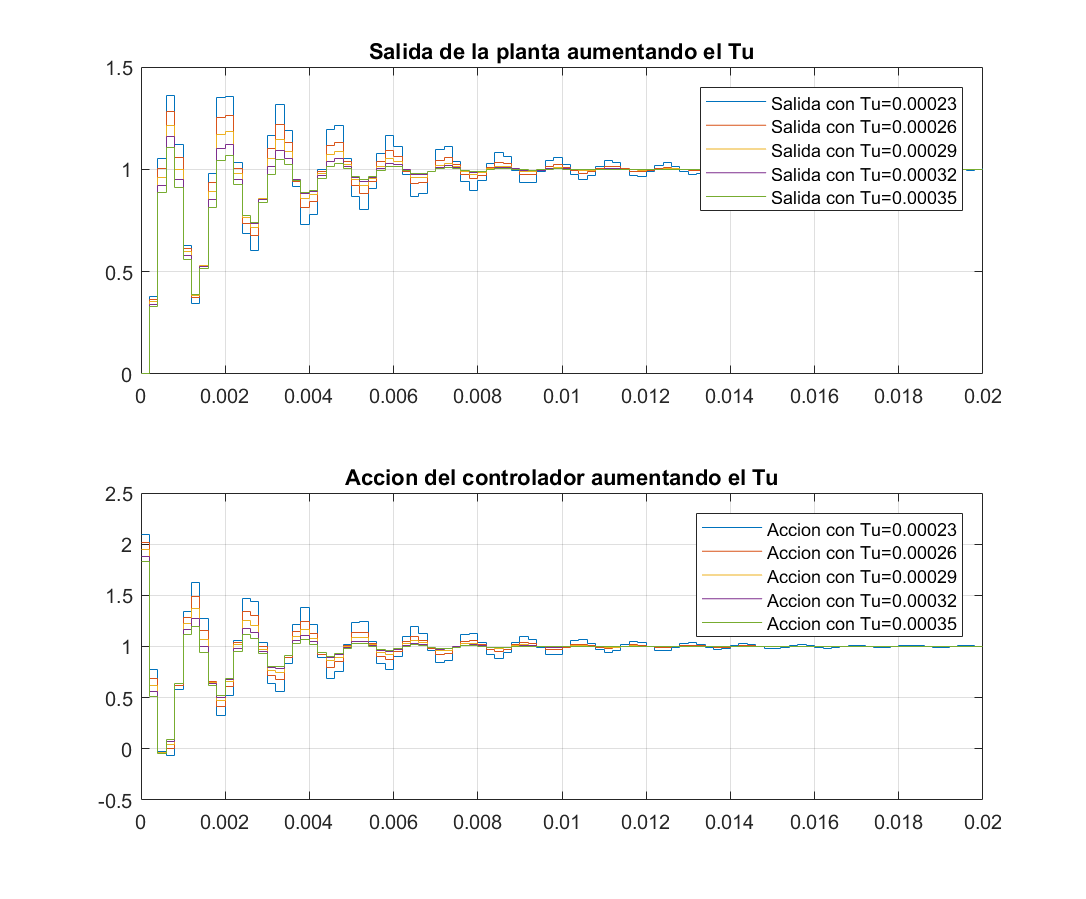
title('Salida de la planta aumentando el Tu');

legend(Legend{:,1});grid on; %agrega todos los nombres de las señales

subplot(2,1,2);

title('Accion del controlador aumentando el Tu');

legend(Legend{:,2});grid on;



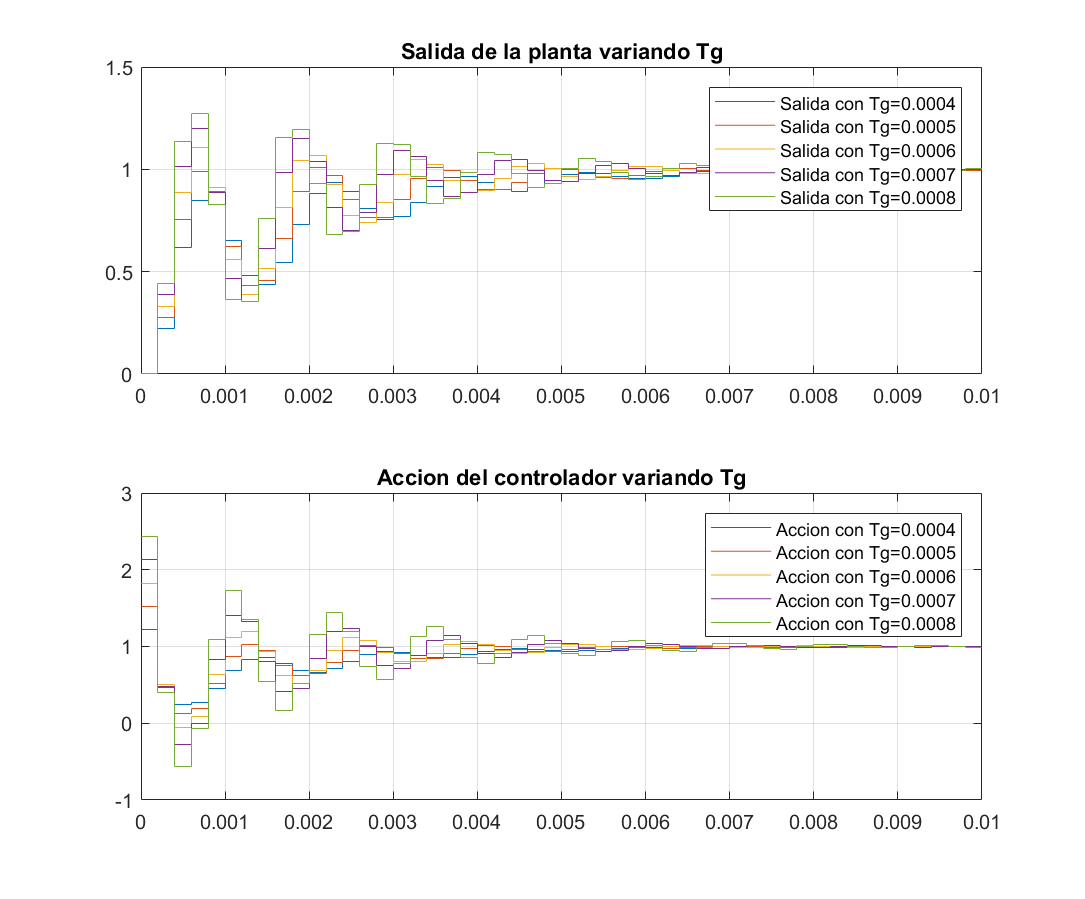
Se observa que aumentando el Tu, el sistema reduce considerablemente las amplitudes de los picos de la respuesta y de la acción de control. En particular, los valores de cada ganancia para Tu=0.00035 son:

kp = 0.8889

ki = 1.1111e+03 / Ki\*T0=0.2222

kd = 1.8750e-04 / Kd/T0=0.9375

Comparandolos con los valores originales (Kp=1.25, Ki=0.5 ,Kd=0.9375), se observa que Kp y Ki disminuyeron considerablemente, no asi Kd. Se repite el experimento anterior tomando ahora Tu=0.00035 t variando Tg. Se utiliza el algoritmo anterior, cambiando solo los títulos de las graficas y variando Tg en vez de Tu. La salida obtenida es la siguiente:



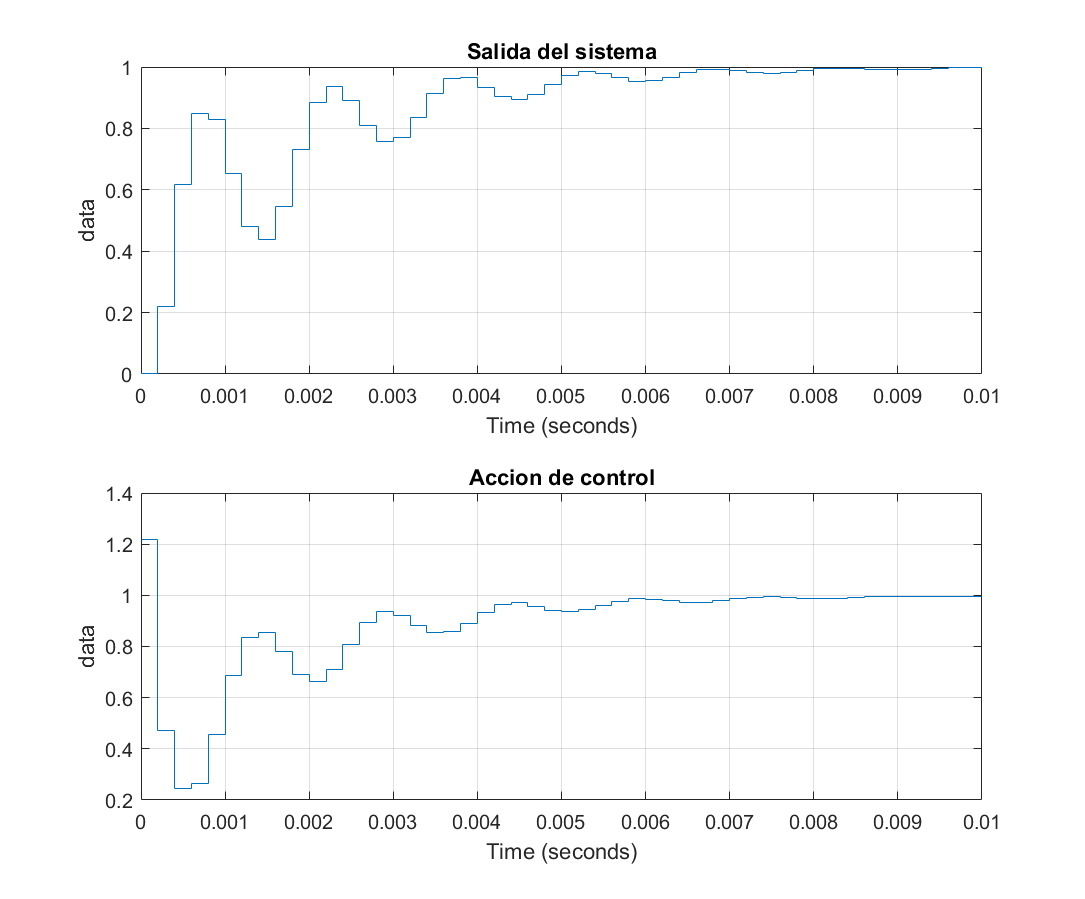
Se observa que los mejores resultados se obtienen con un Tg de 0.0004. Con este valor de Tg, los valores de las ganancias son los siguientes:

kp = 0.5926

ki = 740.7407 / ki\*T0=0.1481

kd = 1.2500e-04 / kd/T0=0.6250

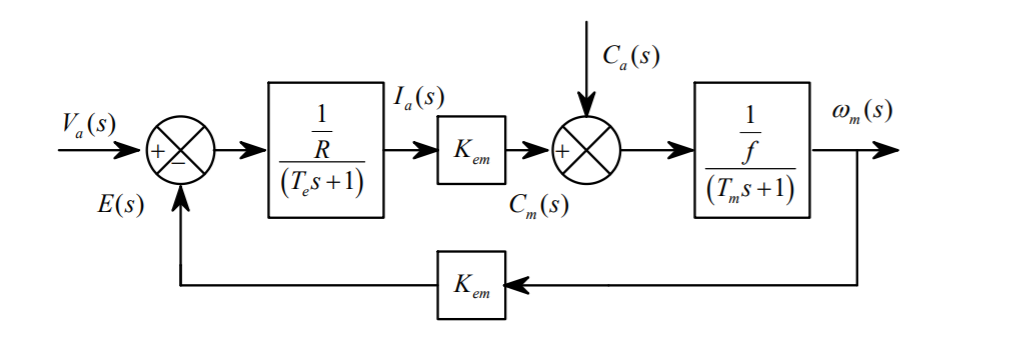
Graficando solo esta función, los resultados son los siguientes:

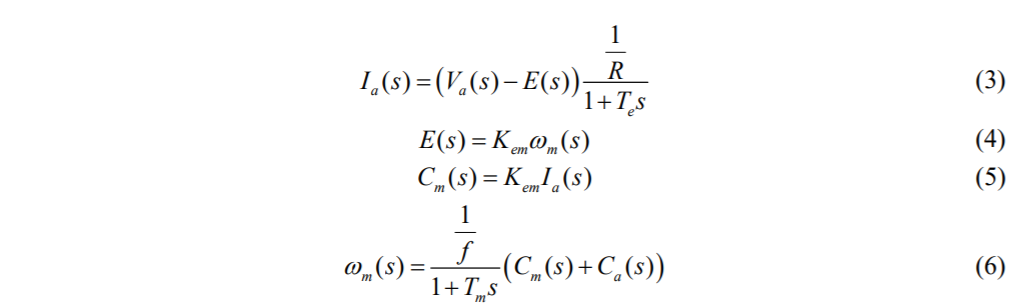


Se observa que el tiempo de establecimiento es mucho más rápido que los anteriores, no hay sobreimpulsos y la acción de control toma valores muy acotados.

# MOTOR DE DC DEL TP2. PID convencional vs PID modificado.

Modelo de un motor de corriente continua expresado en las Ecs. (3), (4), (5) y (6) y esquematizado en el diagrama de bloques presentado en la Fig. (3)





La alimentación del motor es una tensión continua de 100 V, la carga mecánica es un par antagónico de 1 Nt aplicada a los 0.15 s del arranque.

## Determine el período de muestreo y justifique la elección del mismo.

Se cargan las constantes en matlab y se genera la función de transferencia:

%% punto 3

R=8;L=.08;Te=L/R;Kem=0.67;

J=2.22\*10^-3;f=1.86\*10^-3;Tm=J/f;

Va=100;Tcarga=0.15;

G1=tf(1/R,[Te 1]);

G2=tf(1/f,[Tm 1]);

H=feedback(G1\*G2\*Kem,Kem)

[n,d]=tfdata(H,'v');

p=pole(H)

La función de transferencia del motor para la entrada de referencia es la siguiente:

|  |  |  |
| --- | --- | --- |
|  |  | (9) |

Y sus polos están ubicados en:

p=

-50.4189 + 8.3251i

-50.4189 - 8.3251i

Por lo tanto, su constante temporal es 1/50=20ms. Se selecciona un periodo de muestreo 5 veces menor, por lo tanto T0=4ms.

## Emplee Ziegler-Nichols de lazo cerrado para ajustar el PID convencional y Takahashi para el modificado. Muestre los resultados. Ajuste por prueba y error a la mejor respuesta posible para cada uno.

## Presente una comparación entre ambos.

# INDICE DE DESEMPEÑO.

Para alguno de los controladores PID ajustados, programa un algoritmo recursivo para encontrar los valores de las constantes tales que minimicen algún índice propuesto por ustedes 𝐽, el cual debe depender al menos del error cuadrático medio. Tomen una variación de 20% para cada uno de los valores ajustados y presenten gráficas de:

## Código .m del algoritmo programado.

clc;close all;

%% punto 4

n=1;d=[1 3 2]; %se cargan los valores de num y denom

G=tf(n,d) %se arma la funcion de transferencia

T0=.1;

[nd,dd]=c2dm(n,d,T0,'zoh')

gg=tf(nd,dd,T0)

n=0:T0:7;

kp=4;ki=2;kd=2;

%hasta este punto, carga los datos previos

sim("punto1.slx"); %grafica la planta original

figure();

subplot(2,1,1);

plot(Salida);

subplot(2,1,2);

plot(Accioncontrol);

r=.3; %Coeficiente que determina relacion entre

%el menor error y la menor variacion de la accion

coef\_menor=10000; %Valor grande para ser sobreescrito

cant\_iter=10; %Define la cantidad de valores que prueba en cada K

porcentaje=.2; %Define el rango en que varia

min=1-porcentaje; %cada ganancia se varia un +-70%

max=1+porcentaje; %Ya que al variar solo un +-20%,

i=kp\*min:(max-min)\*kp/cant\_iter:kp\*max+.1; %Kp tomaba el menor

j=ki\*min:(max-min)\*ki/cant\_iter:ki\*max+.1; %valor. La idea fue

k=kd\*min:(max-min)\*kd/cant\_iter:kd\*max+.1; %no limitar los valores

c=[]; %c guarda todos los valores de los coeficientes

for kp=i %evalua todo el rango de valores

for ki=j %para las tres variables

for kd=k

sim('punto1.slx');

%el coeficiente utilizado realiza la sumatoria de

%cada valor del error al cuadrado mas la desviacion

%de la accion de control respecto al promedio al cuadrado

%ponderada por un indice "r"

accion=Accioncontrol.Data; %para instrucciones mas cortas

error=sum((Salida.Data-1).^2);

%el error es la salida menos la entrada,

%al ser un escalon su valor siempre es 1

delta\_accion=sum((accion-mean(accion)).^2);

coef=error+delta\_accion\*r; %obtiene el coeficiente

c=[c;coef kp ki kd];

if coef<coef\_menor %Si es menor

coef\_menor=coef %que el menor anterior

kpf=kp; %guarda todas las variables

kif=ki;

kdf=kd;

end

% error=0;accion=0;

% for l=1:length(Salida.Data)

%

% end

end

end

end

kd=kdf; ki=kif; kp=kpf; %finalmente se reemplazan las variables

sim("punto1.slx"); %por los valores anteriores y se vuelve a

subplot(2,1,1);hold on; %graficar.

plot(Salida);

grid;title("Salida del sistema");

legend("Salida original","Salida optimizada con r="+r);

subplot(2,1,2); hold on;

plot(Accioncontrol);

grid;title("Accion de control");

legend("Accion original","Accion optimizada con r="+r);

## Variación del índice para todos los casos.

Al tomarse 10 valores para cada índice, son un total de 1000 resultados.

Columns 1 through 8 16.0390 16.8520 17.7080 18.6062 19.5458 20.5262 15.8821 16.6819

Columns 9 through 16 17.5256 18.4125 19.3416 20.3123 15.8109 16.5965 17.4272 18.3020

Columns 17 through 24 19.2199 20.1801 15.8146 16.5853 17.4022 18.2642 19.1702 20.1194

Columns 25 through 32 15.8847 16.6398 17.4422 18.2907 19.1842 20.1218 16.0144 16.7529

Columns 33 through 40 17.5401 18.3745 19.2549 20.1803 18.0977 18.9901 19.9246 20.9005

Columns 41 through 48 21.9171 22.9738 17.9638 18.8443 19.7677 20.7333 21.7403 22.7882

Columns 49 through 56 17.9098 18.7776 19.6893 20.6439 21.6408 22.6792 17.9261 18.7804

Columns 57 through 64 19.6796 20.6227 21.6088 22.6372 18.0046 18.8448 19.7309 20.6618

Columns 65 through 72 21.6366 22.6545 18.1391 18.9645 19.8368 20.7549 21.7178 22.7246

Columns 73 through 80 20.3704 21.3405 22.3519 23.4040 24.4963 25.6283 20.2588 21.2180

Columns 81 through 88 22.2192 23.2620 24.3456 25.4695 20.2220 21.1697 22.1603 23.1931

Columns 89 through 96 24.2674 25.3827 20.2512 21.1867 22.1660 23.1883 24.2528 25.3590

Columns 97 through 104 20.3391 21.2618 22.2292 23.2404 24.2946 25.3912 20.4798 21.3891

Columns 105 through 112 22.3439 23.3435 24.3869 25.4734 22.8482 23.8945 24.9816 26.1090

Columns 113 through 120 27.2762 28.4828 22.7580 23.7945 24.8724 25.9912 27.1505 28.3496

Columns 121 through 128 22.7385 23.7644 24.8325 25.9423 27.0930 28.2843 22.7811 23.7959

Columns 129 through 136 24.8537 25.9538 27.0956 28.2786 22.8791 23.8823 24.9292 26.0192

Columns 137 through 144 27.1516 28.3257 23.0272 24.0181 25.0536 26.1330 27.2554 28.4203

Columns 145 through 152 25.5241 26.6458 27.8080 29.0101 30.2517 31.5325 25.4550 26.5677

Columns 153 through 160 27.7214 28.9156 30.1499 31.4239 25.4525 26.5556 27.7003 28.8862

Columns 161 through 168 30.1127 31.3795 25.5089 26.6018 27.7370 28.9141 30.1324 31.3914

Columns 169 through 176 25.6178 26.7000 27.8253 28.9930 30.2026 31.4536 25.7741 26.8451

Columns 177 through 184 27.9600 29.1179 30.3185 31.5610 28.3933 29.5899 30.8267 32.1031

Columns 185 through 192 33.4190 34.7739 28.3449 29.5332 30.7621 32.0313 33.3404 34.6891

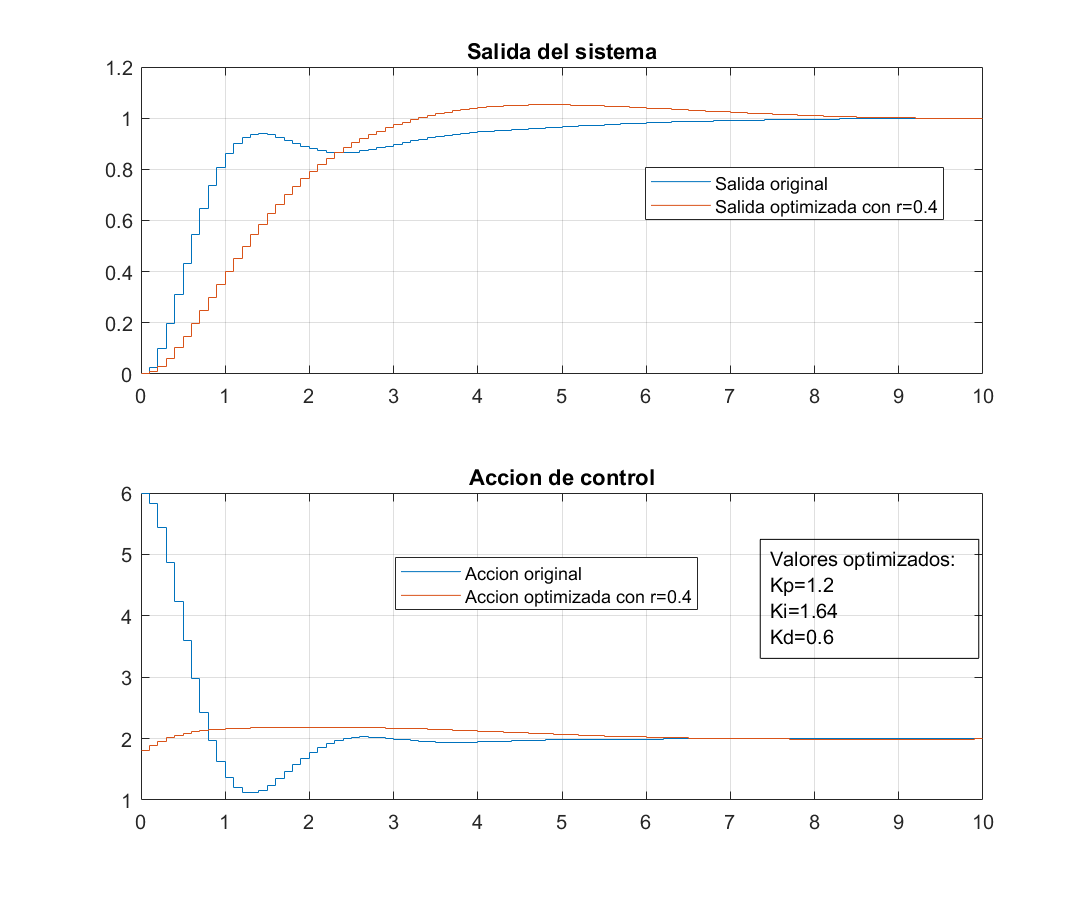
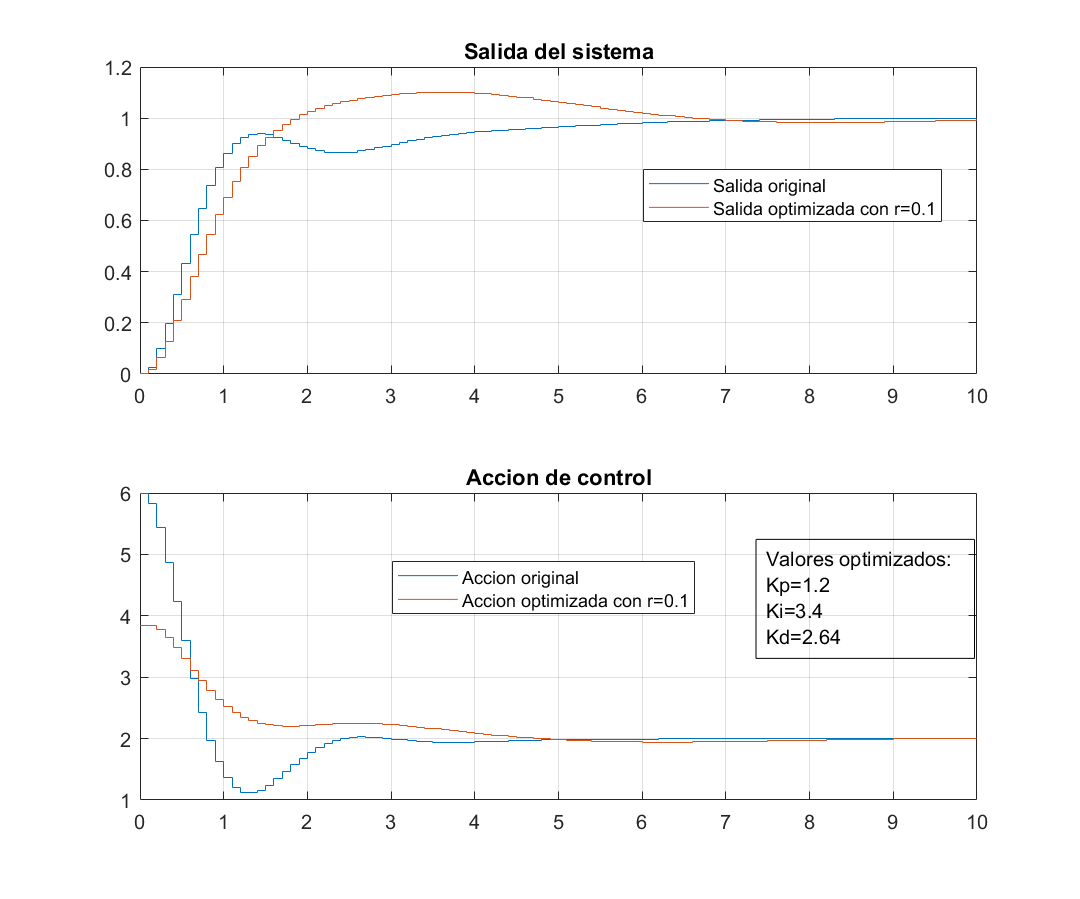
Columns 193 through 200 28.3595 29.5389 30.7596 32.0211 33.3230 34.6650 28.4298 29.5999

Columns 201 through 208 30.8119 32.0653 33.3596 34.6945 28.5502 29.7104 30.9132 32.1581

Columns 209 through 216 33.4444 34.7718 28.7157 29.8656 31.0588 32.2946 33.5726 34.8922

## Comparación de la respuesta con mejor índice vs la respuesta obtenida en el ejercicio elegido

Se corrió el algoritmo varias veces variando el parámetro “r”, con un porcentaje de variación de +-70% y mas muestras por cada valor, ya que se observaba que para observar la influencia de las diferentes ponderaciones en los valores finales de las tres ganancias. A continuación se reproducen los distintos resultados:



Se observa que al aumentar el parámetro r (que implica que la variación de la acción de control tiene un peso mayor en el resultado del coeficiente), la acción de control fluctua menos, a costa de un tiempo de levantamiento más lento. Ademas, se probó aumentar el r a valores de 0.7 y 1, pero se observo que el único cambio se daba en el Ki, variando a 1.62 y 1.56 respectivamente, por lo tanto no se reprodujeron las graficas en el informe.